



UNIVERSITA' DEGLI STUDI DI PARMA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Fondamenti di Informatica B

Lezione n.3

Fondamenti di Informatica B

Lezione n.3

- Forme canoniche
- Trasformazioni
- Esercizi

In questa lezione verranno considerate le proprietà dell'algebra booleana che saranno poi utili per l'analisi e la progettazione di circuiti a livello logico

Introdurremo poi le tecniche per trasformare la rappresentazione algebrica di un problema nella rappresentazione circuitale

Forme canoniche

- Un minterm è un'espressione prodotto che contiene in modo affermato o negato tutte le variabili della funzione

– es: in una funzione di tre variabili $\{x_1, x_2, x_3\}$ sono minterm le seguenti espressioni:

$$\overline{x_1} \bullet \overline{x_2} \bullet x_3 \quad x_1 \bullet \overline{x_2} \bullet x_3$$

– mentre non sono minterm:

$$\overline{x_1} \bullet \overline{x_2} \quad x_1 \bullet x_3$$

Forme canoniche

- Un maxterm è un'espressione somma che contiene in modo affermato o negato tutte le variabili della funzione

– es: in una funzione di tre variabili $\{x_1, x_2, x_3\}$ sono maxterm espressioni quali:

$$\overline{x_1} + \overline{x_2} + x_3 \quad x_1 + \overline{x_2} + x_3$$

– mentre non lo sono:

$$\overline{x_1} + \overline{x_2} \quad x_1 + x_3$$

Forme canoniche

- Si dicono forme canoniche le somme di minterm (SdP) e i prodotti di maxterm (PdS)

– esempi di forme canoniche:

- SdP: $(\overline{x_1} \bullet \overline{x_2} \bullet x_3) + (x_1 \bullet \overline{x_2} \bullet x_3)$

- PdS: $(\overline{x_1} + \overline{x_2} + x_3)(x_1 + \overline{x_2} + x_3)$

La somma di numeri binari

■ Si consideri la somma tra:

$$\begin{array}{r} \text{riporto} \quad 1 \ 1 \ 1 \\ \text{addendo} \quad 1 \ 0 \ 1 \ 1 \ + \\ \text{addendo} \quad 0 \ 1 \ 1 \ 1 \ = \\ \hline \text{somma} \quad 1 \ 0 \ 0 \ 1 \ 0 \end{array}$$

S_0 e C_1 sono funzioni
booleane nelle variabili
 $\{x_0, y_0, C_0\}$

x_0	y_0	C_0	S_0	C_1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

La somma di numeri binari

- Per sommare due numeri binari è quindi necessario sintetizzare due funzioni:
 - $S_0(x_0, y_0, C_0)$
 - $C_1(x_0, y_0, C_0)$
- Per sintetizzare una funzione è possibile ricorrere a:
 - somma di minterm (forma SdP)
 - prodotto di maxterm (forma PdS)

La somma di numeri binari

■ Esempio: sintetizzare la funzione C_1

- Selezionare gli 1 della funzione
- a ciascuno di essi corrisponde un minterm

$$C_1 = \overline{x_0}y_0C_0 + x_0\overline{y_0}C_0 + x_0y_0\overline{C_0} + x_0y_0C_0$$

Una somma di minterm rappresenta direttamente tutti gli 1 di una funzione

x_0	y_0	C_0	S_0	C_1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

La somma di numeri binari

■ Esempio: sintetizzare la funzione C_1

- Selezionare gli 0 della funzione
- a ciascuno di essi corrisponde un maxterm

$$C_1 = (x_0 + y_0 + C_0) \cdot (x_0 + y_0 + \overline{C_0}) \cdot (x_0 + \overline{y_0} + C_0) \cdot (\overline{x_0} + y_0 + C_0)$$

Un prodotto di maxterm rappresenta direttamente tutti gli 0 di una funzione

x_0	y_0	C_0	S_0	C_1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Forme canoniche

- Una qualunque funzione $f(x_1, x_2, \dots, x_n)$ può essere espressa univocamente da una forma canonica costituita da:

- somma di minterm:

$$f(x_1, x_2, \dots, x_n) = \sum_i (x_{i,1} \bullet x_{i,2} \bullet \dots \bullet x_{i,n})$$

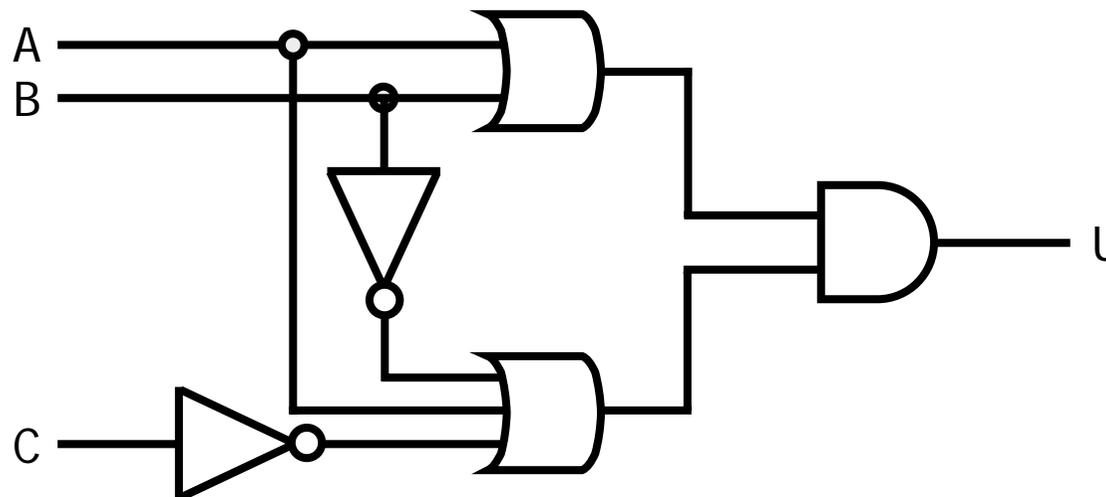
- prodotto di maxterm:

$$f(x_1, x_2, \dots, x_n) = \prod_i (x_{i,1} + x_{i,2} + \dots + x_{i,n})$$

Logica a due livelli

- Qualunque espressione del tipo prodotto di somme (PdS) può essere implementata mediante un circuito a due livelli di logica

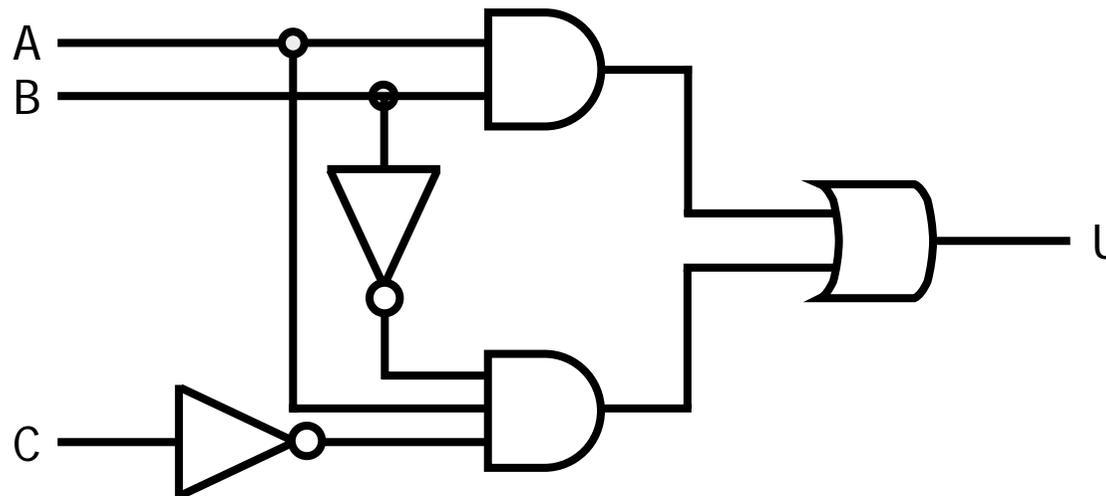
– es: $U = (A + B)(A + \overline{B} + \overline{C})$



Logica a due livelli

- Qualunque espressione del tipo somma di prodotti (SdP) può essere implementata mediante un circuito a due livelli di logica

– es: $U = (A \bullet B) + (A \bullet \bar{B} \bullet \bar{C})$



Logica a due livelli

- Anche le funzioni S_0 e C_1 possono essere sintetizzate con due livelli di logica

x_0	y_0	C_0	S_0	C_1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$C_{1(SP)} = \overline{x_0}y_0C_0 + x_0\overline{y_0}C_0 + x_0y_0\overline{C_0} + x_0y_0C_0$$

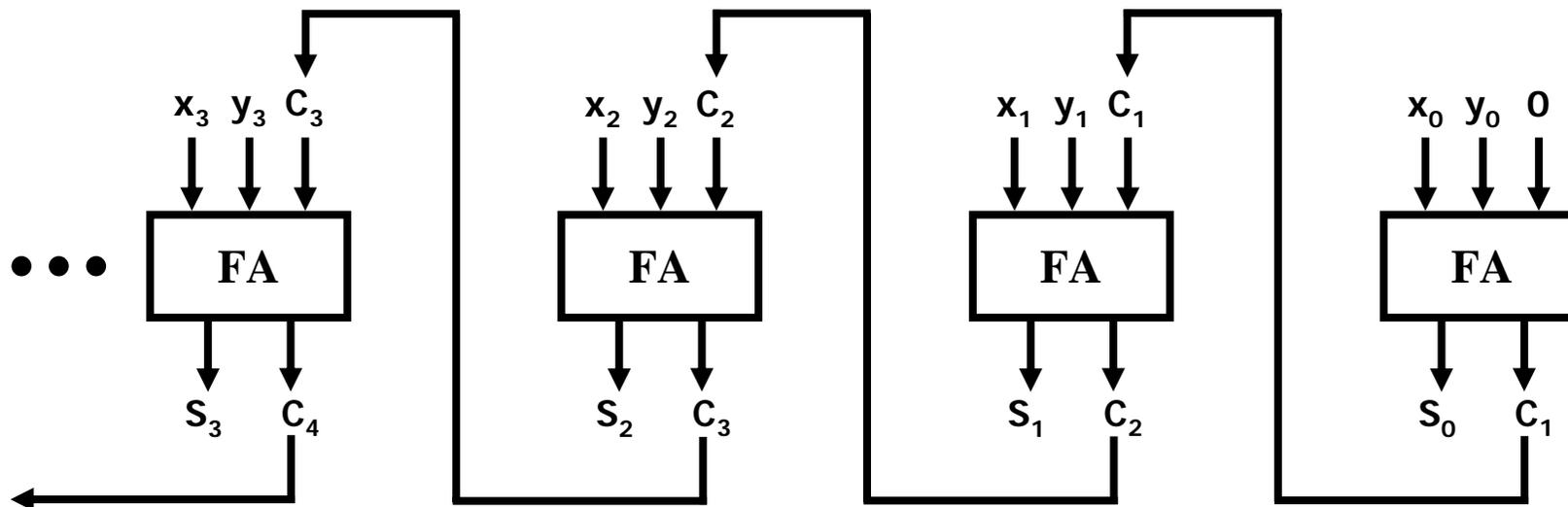
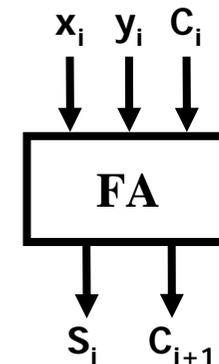
$$C_{1(PS)} = (x_0 + y_0 + C_0)(x_0 + y_0 + \overline{C_0}) \bullet \\ (x_0 + \overline{y_0} + C_0)(\overline{x_0} + y_0 + C_0)$$

$$S_{0(SP)} = \overline{x_0}\overline{y_0}C_0 + \overline{x_0}y_0\overline{C_0} + x_0\overline{y_0}\overline{C_0} + x_0y_0C_0$$

$$S_{0(PS)} = (x_0 + y_0 + C_0)(x_0 + \overline{y_0} + \overline{C_0}) \bullet \\ (\overline{x_0} + y_0 + \overline{C_0})(\overline{x_0} + \overline{y_0} + C_0)$$

Full adder

- Un circuito che implementa le funzioni S_0 e C_1 è detto Full Adder (FA) o sommatore completo
- Connettendo più FA assieme è possibile realizzare un circuito che somma numeri binari di lunghezza qualsiasi



Negazione logica

- Un cerchio all'ingresso o all'uscita di una porta logica significa negazione
- I nuovi simboli così costruiti possono essere utilizzati per rappresentare funzioni booleane anche complesse

– es: $U = \overline{A} + \overline{B}$



Equivalenze

- Il Teorema di De Morgan afferma che:

$$\overline{A \bullet B} = \overline{A} + \overline{B}$$

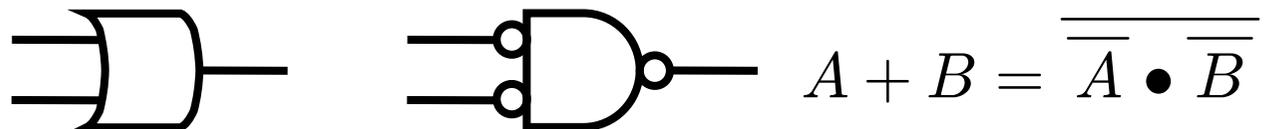
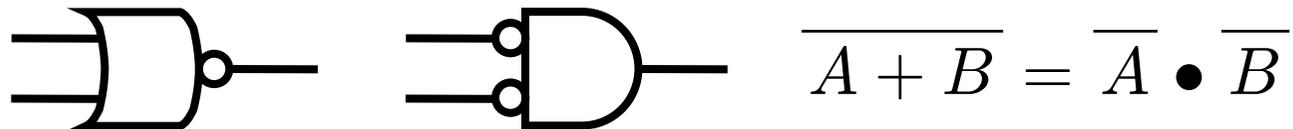
che corrisponde all'equivalenza circuitale:



Le relazioni di equivalenza dell'algebra booleana si riflettono, a livello circuitale, in relazioni di equivalenza fra moduli logici

Equivalenze

- E' possibile effettuare trasformazioni circuitali basate su proprietà logiche:

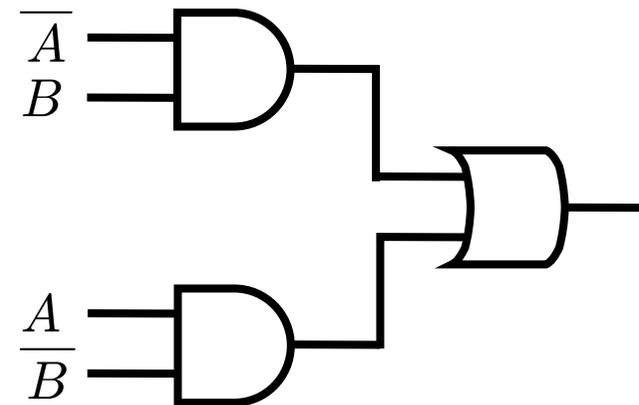


La stessa funzione logica può essere rappresentata da circuiti diversi

Trasformazioni circuitali

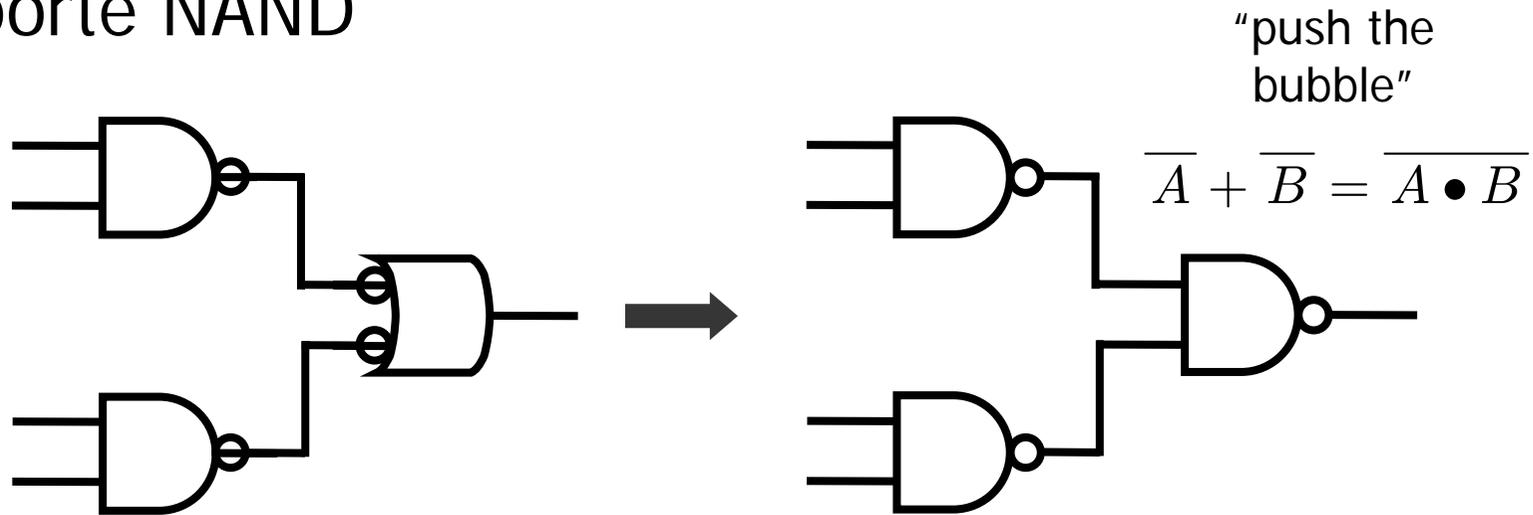
- Funzione ex-or realizzata con un circuito a due livelli di logica

	A	B	U
	0	0	0
$\overline{A}B$	0	1	1
$A\overline{B}$	1	0	1
	1	1	0



Trasformazioni circuitali

- Aggiungendo due negazioni all'inizio e alla fine della linea di un segnale, la funzione sintetizzata non cambia
- Sfruttando una delle equivalenze logiche, è possibile ottenere la stessa funzione con sole porte NAND



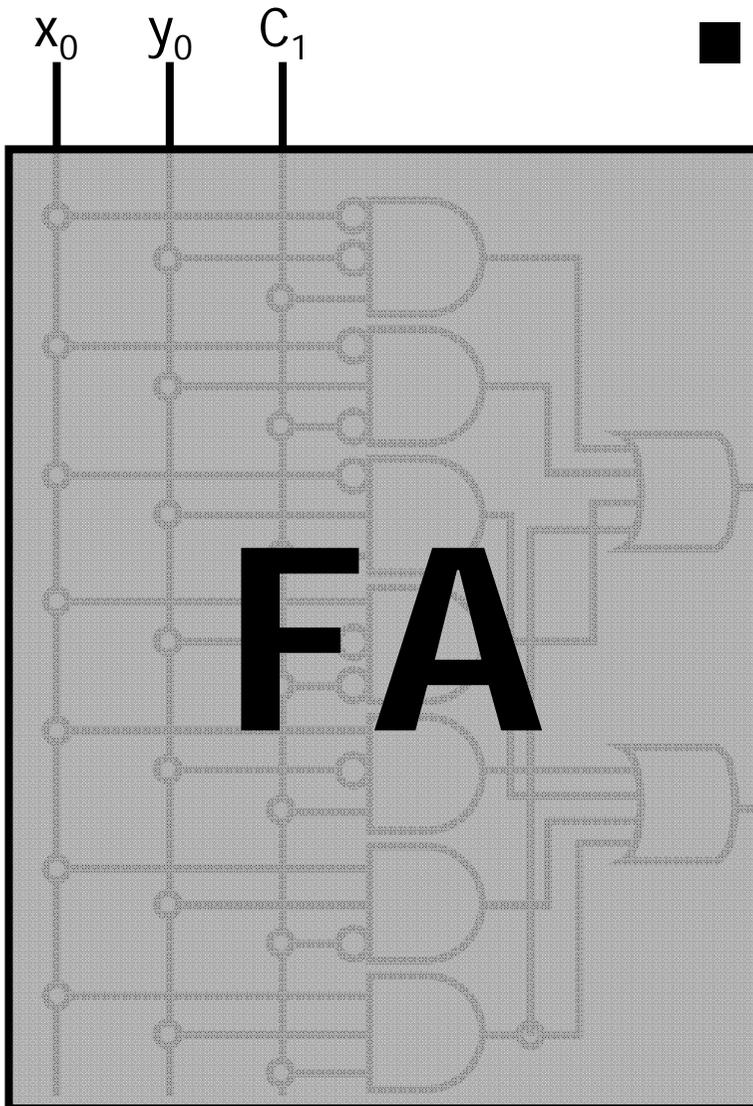
Trasformazioni circuitali

- Trasformazioni analoghe alla precedente possono essere applicate in casi più complessi:

$$\begin{aligned}ABC + DEF &= \overline{\overline{ABC} \bullet \overline{DEF}} \\(A + B + C)(D + E + F) &= \overline{\overline{A + B + C} + \overline{D + E + F}}\end{aligned}$$

- E' possibile sintetizzare qualunque funzione rappresentata come SdP o PdS utilizzando solo porte NAND o solo porte NOR
 - sia le porte NAND che le porte NOR sono insiemi completi

Realizzazione circuitale di un FA



- Circuito a due livelli (SdP) che sintetizza le funzioni logiche necessarie per realizzare un FA:

$$S_{0(SP)} = \overline{x_0} \overline{y_0} C_0 + \overline{x_0} y_0 \overline{C_0} + x_0 \overline{y_0} \overline{C_0} + x_0 y_0 C_0$$

$$C_{1(SP)} = \overline{x_0} y_0 C_0 + x_0 \overline{y_0} C_0 + x_0 y_0 \overline{C_0} + x_0 y_0 C_0$$

Esercizi

- Verificare la seguente identità:

$$AB + \overline{A}C + BC = AB + \overline{A}C$$

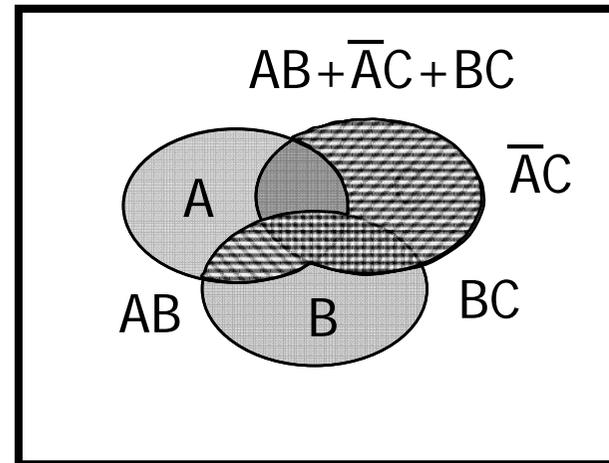
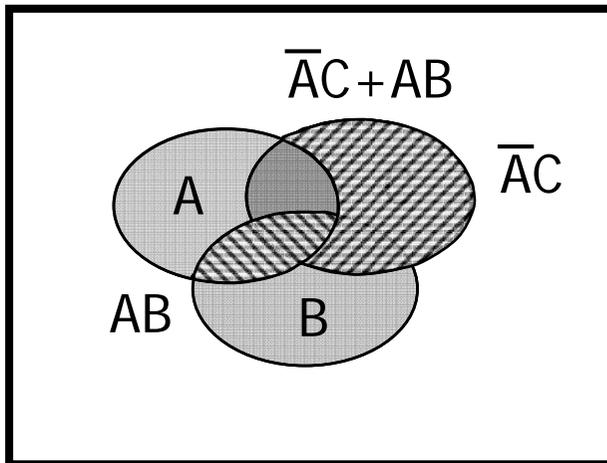
Dimostrazione:

$$\begin{aligned} AB + \overline{A}C + BC &= AB + \overline{A}C + BC(A + \overline{A}) \\ &= AB + ABC + \overline{A}C + \overline{A}BC \\ &= AB(1 + C) + \overline{A}C(1 + B) \\ &= AB + \overline{A}C \end{aligned}$$

- Altre tecniche:
 - confronto delle tabelle della verità
 - diagrammi di Venn
 - confronto delle forme canoniche

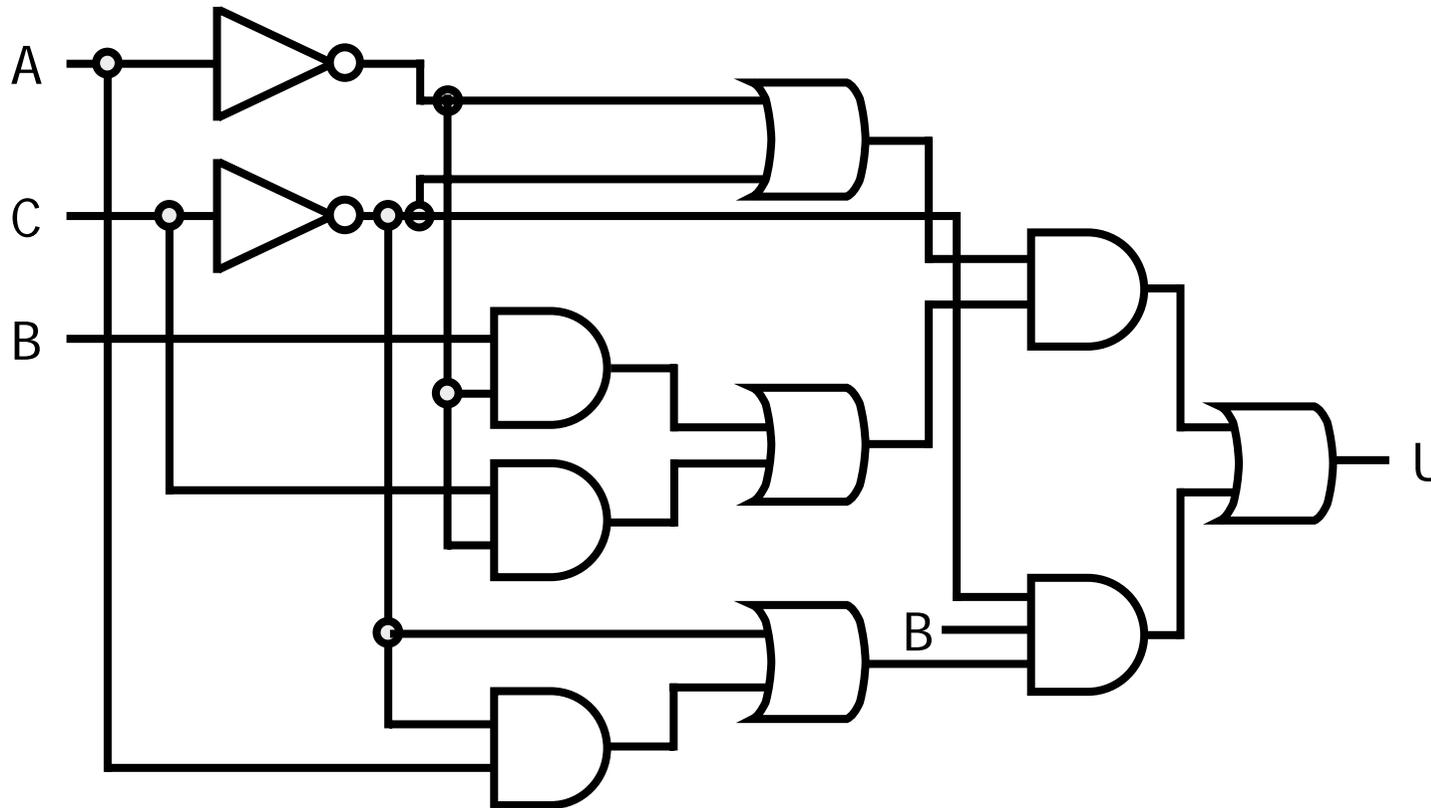
Esercizi

- Verifica dell'equivalenza con i diagrammi di Venn:



Esercizi

■ Semplificare il seguente circuito:



$$U = B\bar{C}(\bar{C} + \bar{C}A) + (\bar{A} + \bar{C})(\bar{A}B + \bar{A}C)$$

Esercizi

■ Occorre semplificare la relazione:

$$\begin{aligned}U &= B\bar{C}(\bar{C} + A\bar{C}) + (\bar{A} + \bar{C})(\bar{A}B + \bar{A}C) \\ &= B\bar{C} + (\bar{A} + \bar{C})(\bar{A}B + \bar{A}C) \\ &= B\bar{C} + (\bar{A} + \bar{C})\bar{A}(B + C) \\ &= B\bar{C} + (\bar{A} + \bar{A}\bar{C})(B + C) \\ &= B\bar{C} + \bar{A}(1 + \bar{C})(B + C) \\ &= B\bar{C} + \bar{A}(B + C)\end{aligned}$$

